
Con riferimento al caso di studio REBU presentato durante il corso, si consideri la seguente variante. Al fine di estendere il servizio, REBU attiva un programma di condivisione delle auto. Un autista durante le ore di riposo può decidere di lasciare la sua auto a disposizione di autisti che possono lavorare ma non hanno un'auto di proprietà con gli standard richiesti da REBU. A tal fine deve indicare la posizione dell'auto e l'intervallo orario in cui non la userà.

Un autista che voglia usare un'auto condivisa cerca tra quelle disponibili, la prenota per il tempo necessario a raggiungerla (le prenotazioni durano max 20 minuti), ne prende possesso, lo segnala. Quando termina il turno la parcheggia in un raggio di 500 metri rispetto a dove l'ha trovata e segnala la nuova posizione.

Il meccanismo di apertura e messa in moto di auto condivise si basa sull'uso di un codice monouso inviato in risposta a una prenotazione.

Domanda 1. Si vuole aggiungere questa nuova funzionalità senza riscrivere tutto il codice esistente.

A. Come si chiama questo principio di progettazione?

Si assuma che non tutte le auto saranno condivise e che ci saranno i due tipi: privata e condivisibile. Si forniscano:

B. il frammento del vecchio diagramma delle classi che descrive il dominio, restringendosi all'autista e alle auto (in cui un autista svolge solo servizio con la propria auto).

C. il frammento modificato.

Domanda 2. Dare un diagramma dei casi d'uso di REBU che comprenda i due nuovi casi d'uso “messa a disposizione di auto” e “presa in uso di auto condivisa”.

Domanda 3. Dare un diagramma di sequenza che descriva la narrativa di “presa in uso di auto condivisa”.

Domanda 4. La progettazione architetturale ha portato alla decisione di usare una factory per la creazione di auto. Dare il frammento di diagramma delle classi che mostra le classi di progettazione coinvolte.

Domanda 5. Si consideri il seguente frammento di codice, che ha lo scopo di controllare se un intervallo temporale A sia interamente contenuto all'interno di un intervallo temporale B (usato dal sistema REBU per controllare se una richiesta di disponibilità di auto condivisibile è compatibile con l'orario di disponibilità di una particolare auto messa in condivisione):

```
public boolean inside(Timespan a, Timespan b)
{
    if (a.start < b.start)
        return false;
    if (a.end > b.end)
        return false;
    if ((a.start >= b.start) && (a.end <= b.end))
        return true;
    return false;
}
```

Si adotti un atteggiamento di *defensive programming*, ovvero ci si proponga di realizzare una suite di test che consenta di verificare il funzionamento del metodo `inside()` senza fare assunzioni sulla correttezza dei parametri.

Oltre a dare la lista di casi di test, si commenti su quale criterio o insieme di criteri sono stati usati per generarla, e si propongano eventuali correzioni al codice (derivanti dai risultati del test).

Infine, si proponga il voto all'esame di Programmazione I del programmatore che ha scritto il codice.